Créer un compte Se connecter

Article Discussion Lire Modifier le code Afficher l'historique Rechercher

WIKIPÉDIA
L'encyclopédie libre

Navigation
Accueil
Portails thématiques
Article au hasard
Contact

Contribuer
Débuter sur Wikipédia
Aide
Communauté
Modifications récentes
Faire un don

Imprimer / exporter
Créer un livre
Télécharger comme PDF
Version imprimable

Outils
Pages liées
Suivi des pages liées
Importer un fichier
Pages spéciales
Adresse de cette version
Information sur la page
Élément sur Wikidata
Citer cette page

Autres langues
Afrikaans
العربية
Azərbaycanca
Беларуская
Беларуская (тарашкевіца)
Български
বাংলা
Bosanski
Català
Čeština
Cymraeg
Dansk
Deutsch
Zazaki
Ελληνικά
English
Esperanto
Español
Eesti
Euskara
فارسی
Suomi
Võro
Føroyskt

Hypertext Transfer Protocol

(Redirigé depuis HTTP)

L'***HyperText Transfer Protocol***, plus connu sous l'abréviation **HTTP** — littéralement « protocole de transfert hypertexte » — est un protocole de communication client-serveur développé pour le World Wide Web. HTTPS (avec S pour *secured*, soit « sécurisé ») est la variante du HTTP *sécurisée* par l'usage des protocoles SSL ou TLS.

Hypertext Transfer Protocol	
Fonction	Transmission d'hypertexte
Sigle	HTTP
Date de création	1990
Port	80
RFC	1996 : RFC 1945 1997 : RFC 2068 1999 : RFC 2616

modifier

HTTP est un protocole de la couche application. Il peut fonctionner sur n'importe quelle connexion fiable, dans les faits on utilise le protocole TCP comme couche de transport. Un serveur HTTP utilise alors par défaut le port 80 (443 pour HTTPS).

Les clients HTTP les plus connus sont les navigateurs Web permettant à un utilisateur d'accéder à un serveur contenant les données. Il existe aussi des systèmes pour récupérer automatiquement le contenu d'un site tel que les aspirateurs de site Web ou les robots d'indexation.

Ces clients se connectent à des serveurs HTTP tels qu'Apache HTTP Server ou Internet Information Services.

Pile de protocoles

7. Application
6. Présentation
5. Session
4. Transport
3. Réseau
2. Liaison
1. Physique

Modèle Internet
Modèle OSI

modifier

Sommaire

1 Historique
2 Méthodes
3 Du client au serveur
4 Identification
5 HTTP 0.9
6 HTTP 1.0
7 HTTP 1.1
8 HTTP 2.0
9 Liste de serveurs HTTP
10 Notes et références
11 Voir aussi
11.1 Articles connexes
11.2 Liens externes

Historique [modifier le code]

Article détaillé : World Wide Web#Historique.

HTTP a été inventé par Tim Berners-Lee avec les adresses Web et le langage HTML pour créer le World Wide Web. À cette époque, le *File Transfer Protocol* (FTP) était déjà disponible pour transférer des fichiers, mais il ne supportait pas la notion de format de données telle qu'introduite par *Multipurpose Internet Mail Extensions* (MIME). La première version de HTTP était très élémentaire, mais prévoyait déjà le support d'en-têtes MIME pour décrire les données transmises. Cette première version reste encore partiellement utilisable de nos jours, connue sous le nom de HTTP/0.9.

En mai 1996, HTTP/1.0 voit le jour et est décrit dans la RFC 1945 . Cette version supporte les serveurs HTTP virtuels, la gestion de cache et l'identification.

En janvier 1997, HTTP/1.1 devient finalement standard de l'IETF. Il est décrit dans la RFC 2068 de l'IETF, puis dans la RFC 2616 en juin 1999. Cette version ajoute le support du transfert en pipeline (ou pipelinage) et la négociation de type de contenu (format de données, langue).

Méthodes [modifier le code]

Gaeilge
Galego
עברית
Hrvatski
Magyar
Bahasa Indonesia
Íslenska
Italiano
日本語
Қазақша
한국어
Lëtzebuergesch
Lietuvių
Latviešu
Олык марий
Македонски
മലയാളം
Bahasa Melayu
नेपाल भाषा
Nederlands
Norsk nynorsk
Norsk bokmål
Occitan
Polski
Português
Română
Русский
Scots
Srpskohrvatski / српскохрватски
Simple English
Slovenčina
Slovenščina
Shqip
Српски / srpski
Svenska
தமிழ்
తెలుగు
Тоҷикӣ
ไทย
Tagalog
Türkçe
Українська
Tiếng Việt
Winaray
Yorùbá
中文
粵語
Modifier les liens

Dans le protocole HTTP, une méthode est une **Commande** spécifiant un type de requête, c'est-à-dire qu'elle demande au serveur d'effectuer une action. En général l'action concerne une ressource identifiée par l'URL qui suit le nom de la méthode.

Dans l'illustration ci-contre, une requête GET est envoyée pour récupérer la page d'accueil du site web www.perdu.com :

```
GET / HTTP/1.1
Host: www.perdu.com
```

Capture d'écran d'une requête GET et de sa réponse.

Il existe de nombreuses méthodes, les plus courantes étant GET, HEAD et POST :

GET
: C'est la méthode la plus courante pour demander une ressource. Une requête GET est sans effet sur la ressource, il doit être possible de répéter la requête sans effet.

HEAD
: Cette méthode ne demande que des informations sur la ressource, sans demander la ressource elle-même.

POST
: Cette méthode est utilisée pour transmettre des données en vue d'un traitement à une ressource (le plus souvent depuis un formulaire HTML). L'URI fournie est l'URI d'une ressource à laquelle s'appliqueront les données envoyées. Le résultat peut être la création de nouvelles ressources ou la modification de ressources existantes. À cause de la mauvaise implémentation des méthodes HTTP (pour Ajax) par certains navigateurs (et la norme HTML qui ne supporte que les méthodes GET et POST pour les formulaires), cette méthode est souvent utilisée en remplacement de la requête PUT, qui devrait être utilisée pour la mise à jour de ressources.

OPTIONS
: Cette méthode permet d'obtenir les options de communication d'une ressource ou du serveur en général.

CONNECT
: Cette méthode permet d'utiliser un proxy comme un tunnel de communication.

TRACE
: Cette méthode demande au serveur de retourner ce qu'il a reçu, dans le but de tester et effectuer un diagnostic sur la connexion.

PUT
: Cette méthode permet de remplacer ou d'ajouter une ressource sur le serveur. L'URI fourni est celui de la ressource en question.

PATCH
: cette méthode permet, contrairement à PUT, de faire une modification **partielle** d'une ressource.

DELETE
: Cette méthode permet de supprimer une ressource du serveur.

Ces 3 dernières méthodes nécessitent généralement un accès privilégié.

Certains serveurs autorisent d'autres méthodes de gestion de leurs ressources (par exemple WebDAV).

Du client au serveur [modifier le code]

La liaison entre le client et le serveur n'est pas toujours directe, il peut exister des machines intermédiaires servant de relais :

- Un proxy (ou serveur mandataire) peut modifier les réponses et requêtes qu'il reçoit et peut gérer un cache des ressources demandées.
- Une passerelle (ou *gateway*) est un intermédiaire modifiant le protocole utilisé.
- Un tunnel transmet les requêtes et les réponses sans aucune modification, ni mise en cache.

Identification [modifier le code]

HTTP permet l'identification du visiteur par transmission d'un nom et d'un mot de passe. Il existe 2 modes d'identification : *Basic* et *Digest* (RFC 2617). Le premier mode transmet le mot de passe en clair, et ne doit donc être utilisé qu'avec le protocole HTTPS. Le deuxième mode permet une identification sans transmettre le mot de passe en clair. L'identification est cependant souvent effectuée par une couche applicative supérieure à HTTP.

HTTP 0.9 [modifier le code]

Au début du World Wide Web, il était prévu d'ajouter au protocole HTTP des capacités de négociation de contenu, en s'inspirant notamment de MIME. En attendant, le protocole HTTP 0.9 était extrêmement simple.

1. connexion du client HTTP
2. envoi d'une requête de méthode `GET`
3. réponse du serveur HTTP
4. le serveur ferme la connexion pour signaler la fin de la réponse.

Requête :

```
GET /page.html
```

La méthode `GET` est la seule possible. Le serveur reconnaît qu'il a affaire à une requête HTTP 0.9 au fait que la version n'est pas précisée suite à l'URI.

Réponse :

```
<HTML>
<HEAD>
<TITLE>Exemple</TITLE>
</HEAD>
<BODY>
<P>Ceci est une page d'exemple.</P>
</BODY>
</HTML>
```

Pour répondre à une requête HTTP 0.9, le serveur envoie directement le contenu de la réponse, sans méta-données. Il ne doit jamais se comporter ainsi pour les requêtes HTTP de version supérieure.

Inutile de chercher les versions inférieures à 0.9 du protocole HTTP : elles n'existent pas, car HTTP 0.9 n'avait initialement pas de numéro de version. Il a fallu lui en attribuer un quand HTTP 1.0 est arrivé.

HTTP 1.0 [modifier le code]

Le protocole HTTP 1.0, décrit dans le RFC 1945 , prévoit l'utilisation d'en-têtes spécifiés dans le RFC 822 . La gestion de la connexion reste identique à HTTP 0.9 : le client établit la connexion, envoie une requête, le serveur répond et ferme immédiatement la connexion.

Une requête HTTP présente le format suivant :

```
    Ligne de commande (Commande, URL, Version de protocole)
    En-tête de requête
    [Ligne vide]
    Corps de requête
```

Les réponses HTTP présentent le format suivant :

```
    Ligne de statut (Version, Code-réponse, Texte-réponse)
    En-tête de réponse
    [Ligne vide]
    Corps de réponse
```

Requête :

```
GET /page.html HTTP/1.0
Host: example.com
Referer: http://example.com/
User-Agent: CERN-LineMode/2.15 libwww/2.17b3
```

La version du protocole HTTP est précisée suite à l'URI. La requête doit être terminée par un double retour à la ligne (CRLFCRLF). HTTP 1.0 supporte aussi les méthodes HEAD et POST. On constate l'usage d'en-têtes inspirés de MIME pour transférer les méta-données :

`Host`

Permet de préciser le site web concerné par la requête, ce qui est nécessaire pour un serveur hébergeant plusieurs sites à la même adresse IP (*name based virtual host*, hôte virtuel basé sur le nom). C'est le seul en-tête réellement important.

Referer

Indique l'URI du document qui a donné un lien sur la ressource demandée. Cet en-tête permet aux webmasters d'observer d'où viennent les visiteurs.

User-Agent

Indique le logiciel utilisé pour se connecter. Il s'agit généralement d'un navigateur Web ou d'un robot d'indexation.

Réponse :

```
HTTP/1.0 200 OK
Date: Fri, 31 Dec 1999 23:59:59 GMT
Server: Apache/0.8.4
Content-Type: text/html
Content-Length: 59
Expires: Sat, 01 Jan 2000 00:59:59 GMT
Last-modified: Fri, 09 Aug 1996 14:21:40 GMT

<TITLE>Exemple</TITLE>
<P>Ceci est une page d'exemple.</P>
```

La première ligne donne le code de statut HTTP (200 dans ce cas).

Date

Moment auquel le message est généré.

Server

Indique quel modèle de serveur HTTP répond à la requête.

Content-Type

Indique le type MIME de la ressource.

Content-Length

Indique la taille en octets de la ressource.

Expires

Indique le moment après lequel la ressource devrait être considérée obsolète ; permet aux navigateurs Web de déterminer jusqu'à quand garder la ressource en mémoire cache.

Last-Modified

Indique la date de dernière modification de la ressource demandée.

HTTP 1.1 [modifier le code]

Le protocole HTTP 1.1 est décrit par le RFC 2616 qui rend le RFC 2068 obsolète. La différence avec HTTP 1.0 est une meilleure gestion du cache. L'en-tête **Host** devient obligatoire dans les requêtes.

Les soucis majeurs des deux premières versions du protocole HTTP sont d'une part le nombre important de connexions lors du chargement d'une page complexe (contenant beaucoup d'images ou d'animations) et d'autre part le temps d'ouverture d'une connexion entre client et serveur (l'établissement d'une connexion TCP prend un temps triple de la latence entre client et serveur). Des expérimentations de connexions persistantes ont cependant été effectuées avec HTTP 1.0 (notamment par l'emploi de l'en-tête **Connection: Keep-Alive**), mais cela n'a été définitivement mis au point qu'avec HTTP 1.1.

Par défaut, HTTP 1.1 utilise des connexions persistantes, autrement dit la connexion n'est pas immédiatement fermée après une requête, mais reste disponible pour une nouvelle requête. On appelle souvent cette fonctionnalité *keep-alive*. Il est aussi permis à un client HTTP d'envoyer plusieurs requêtes sur la même connexion sans attendre les réponses. On appelle cette fonctionnalité *pipelining*. La persistance des connexions permet d'accélérer le chargement de pages contenant plusieurs ressources, tout en diminuant la charge du réseau.

La gestion de la persistance d'une connexion est gérée par l'en-tête **Connection**.

HTTP 1.1 supporte la négociation de contenu. Un client HTTP 1.1 peut accompagner la requête pour une ressource d'en-têtes indiquant quels sont les langues et formats de données préférés. Il s'agit des en-têtes dont le nom commence par **Accept-**.

Les en-têtes supplémentaires supportés par HTTP 1.1 sont :

Connection

Cet en-tête peut être envoyé par le client ou le serveur et contient une liste de noms spécifiant les options à utiliser avec la connexion actuelle. Si une option possède des paramètres ceux-ci sont spécifiés par l'en-tête portant le même nom que l'option (**Keep-Alive** par exemple, pour spécifier le nombre maximum de requêtes par connexion). Le nom **close** est réservé pour spécifier que la connexion doit être fermée après traitement de la requête en cours.

Accept

Cet en-tête liste les types MIME de contenu acceptés par le client. Le caractère étoile * peut servir à spécifier tous les types / sous-types.

Accept-Charset

Spécifie les encodages de caractères acceptés.

Accept-Language

Spécifie les langues acceptées.

L'ordre de préférence de chaque option (type, encodage ou langue) est spécifié par le paramètre optionnel **q** contenant une valeur décimale entre 0 (*inacceptable*) et 1 (*acceptable*) inclus (3 décimales maximum après la virgule), valant 1 par défaut.

Le support des connexions persistantes doit également fonctionner dans les cas où la taille de la ressource n'est pas connue d'avance (ressource générée dynamiquement par le serveur, flux externe au serveur, …).

Article détaillé : Chunked transfer encoding.

Pour cela, l'encodage de transfert nommé **chunked** permet de transmettre la ressource par morceaux consécutifs en précédant chacun par une ligne de texte donnant la taille de celui-ci en hexadécimal. Le transfert se termine alors par un morceau de taille nulle, où des en-têtes finaux peuvent être envoyés.

Les en-têtes supplémentaires liés à cet encodage de transfert sont :

Transfer-Encoding

Spécifie l'encodage de transfert. La seule valeur définie par la spécification RFC 2616 est **chunked**.

Trailer

Liste tous les en-têtes figurant après le dernier morceau transféré.

TE

Envoyé par le client pour spécifier les encodages de contenu supportés (**Content-Encoding**, ne pas confondre avec **Transfer-Encoding** car **chunked** est obligatoirement supporté par les clients et serveurs implémentant le standard HTTP/1.1), et spécifie si le client supporte l'en-tête **Trailer** en ajoutant **trailers** à la liste.

HTTP 2.0 [modifier le code]

Une nouvelle version d'HTTP, HTTP 2.0, est en cours de développement au sein du groupe de travail « Hypertext Transfer Protocol Bis » (httpbis) de l'Internet Engineering Task Force[1]. Le développement d'HTTP 2.0 a débuté à la suite de la création du protocole SPDY proposé par Google afin de réduire le temps de chargement des pages Web[2,3]. Le groupe de travail httpbis s'était initialement interdit de proposer une nouvelle version d'HTTP, concentrant son activité sur la clarification des spécifications d'HTTP 1.1. Considérant l'arrivée de SPDY et son adoption rapide sur le Web, avec notamment des implémentations dans deux des principaux navigateurs Web, Google Chrome et Mozilla Firefox, Mark Nottingham, « chair » d'httpbis, a émis l'opinion qu'il était temps d'envisager HTTP 2.0 et proposé d'amender la charte d'httpbis en ce sens, initiant de fait le développement du nouveau protocole[4].

SPDY constituait une option naturelle pour servir de base à HTTP 2.0. Deux autres propositions concurrentes ont été ensuite transmises à l'IETF : le protocole « HTTP Speed+Mobility » par Microsoft[5] et une proposition de mise à jour d'HTTP (« Network-Friendly HTTP Upgrade »)[6]. En juillet 2012, httpbis a publié un appel à expression d'intérêt (« Call for Expression of Interest ») afin de recueillir l'avis d'acteurs du Web sur les propositions[7]. Parmi les réponses obtenues figure celle de Facebook qui a signifié sa préférence pour SPDY[8]. En novembre 2012, l'IETF a publié le premier draft d'HTTP 2.0, qui est une copie directe de SPDY[9].

Liste de serveurs HTTP [modifier le code]

Article détaillé : Serveur HTTP.

- C : Apache, Zeus Web Server, nginx, lighttpd, Cherokee, Hiawatha Webserver
- ASP/ASP.Net(C#, VB.net) : IIS
- Java : Tomcat, Jetty, GlassFish, JBoss, JOnAS, Vert.x
- Python : Zope
- Pike : Caudium
- Ruby : WEBrick, Mongrel, Thin
- Erlang : Yaws
- Javascript : Node.js
- Autres : (en) Comparison of web servers

Notes et références [modifier le code]

1. ↑ (en) « Hypertext Transfer Protocol Bis (httpbis) », sur *datatracker.ietf.org* (consulté le 14 novembre 2013)
2. ↑ (en) « SPDY: An experimental protocol for a faster web » (consulté le 14 novembre 2013)
3. ↑ « SPDY Protocol (Internet draft) », février 2012
4. ↑ (en) Mark Nottingham, « Rechartering HTTPbis », sur *lists.w3.org*, 24 janvier 2012 (consulté le 14 novembre 2013)

5. ↑ (en) « HTTP Speed+Mobility (Internet draft) », 15 juin 2012
6. ↑ (en) « Proposal for a Network-Friendly HTTP Upgrade », 29 mars 2012
7. ↑ (en) « Call for Expressions of Interest », sur *ietf.org*, 3 juillet 2012
8. ↑ (en) « HTTP2 Expression of Interest », 15 juillet 2012
9. ↑ (en) Dio Synodinos, « HTTP 2.0 First Draft Published », sur *InfoQ*, 30 novembre 2012

Voir aussi [modifier le code]

Articles connexes [modifier le code]

- Format ouvert
- Interopérabilité
- Liste des codes HTTP
- Navigateur Web
- Pipelining HTTP
- Protocole de communication
- Serveur HTTP
- World Wide Web
- HTTPS

Liens externes [modifier le code]

- (en) Hypertext Transfer Protocol 1.0 sur le site du W3C
- (en) Hypertext Transfer Protocol 1.1 sur le site de l'IETF
- (en) *Hypertext Transfer Protocol Overview* sur le site du W3C
- (en) Extension LiveHTTPheaders pour Firefox
- (en) RFC 2617, *HTTP Authentification: Basic and Digest Access Authentification*, juin 1999
- (en) *Freeware Protocol Logger* sur HttpTea
- (en) *View Online HTTP Request and Response Header*
- Dossier thématique sur le HTTPS sur cases.lu

v · d · m	Couches du modèle OSI
7. Application	DHCP · DNS · FTP · Gopher · H.323 · **HTTP** · IMAP · IRC · NFS · NNTP · POP3 · RTSP · SILC · SIMPLE · SIP · SMTP · SNMP · SMB-CIFS · SSH · TCAP · Telnet · TFTP · VoIP · XMPP · WebDAV
6. Présentation	AFP · ASCII · ASN.1 · MIME · NCP · SSP · TDI · TLV · Unicode · UUCP · Vidéotex · XDR
5. Session	AppleTalk · DTLS · H.323 · RSerPool · SOCKS · TLS
4. Transport	DCCP · RSVP · RTP · SCTP · SPX · TCP · UDP
3. Réseau	ARP · BGP · Babel · BOOTP · CLNP · ICMP · IGMP · IPv4 · IPv6 · IPX · IS-IS · NetBEUI · OSPF · RARP · RIP · X.25
2. Liaison	Anneau à jeton · ARINC 429 · ATM · AFDX · Bitnet · CAN · Ethernet · FDDI · Frame Relay · HDLC · I²C · IEEE 802.3ad (LACP) · LLC · LocalTalk · MIL-STD-1553 · PPP · STP · Wi-Fi · X.21
1. Physique	4B5B · ADSL · BHDn · Bluetooth · Câble coaxial · Codage bipolaire · CSMA/CA · CSMA/CD · DSSS · E-carrier · EIA-232 · EIA-422 · EIA-449 · EIA-485 · FHSS · IEEE 1394 · HomeRF · IrDA · ISDN · Manchester · Manchester différentiel · Miller · MLT-3 · NRZ · NRZI · NRZM · Paire torsadée · PDH · SDH · SDSL · SONET · T-carrier · USB · VDSL · V.21-V.23 · V.42-V.90 · Wireless USB · 10BASE-T · 10BASE2 · 10BASE5 · 100BASE-TX · 1000BASE-T
Articles liés	Pile de protocoles · Modèle Internet

v · d · m	Schéma d'URI
Officiel	aaa · aaas · acap · cap · cid · crid · data · dav · dict · dns · fax · file · ftp · geo · go · gopher · h323 · **http** · https · im · imap · info · ldap · mailto · mid · news · nfs · nntp · pop · pres · rtsp · sip · sips · snmp · tag · tel · telnet · urn · wais · xmpp
Non-officiel	about · afp · aim · apt · bzr · callto · coffee · cvs · daap · ed2k · feed · fish · gg · git · gizmoproject · iax2 · irc · ircs · itms · ldaps · magnet · mms · msnim · rsync · secondlife · skype · spotify · ssh · svn · sftp · smb · sms · steam · view-source · vzochat · webcal · winamp · wyciwyg · xfire · ymsgr

Portail des réseaux informatiques | Portail de l'informatique
Portail des télécommunications | Portail d'Internet

Catégories : HTTP | Format ouvert | Protocole réseau sur la couche application

Dernière modification de cette page le 29 avril 2014 à 14:18.

Droit d'auteur : les textes sont disponibles sous licence Creative Commons paternité partage à l'identique ; d'autres conditions peuvent s'appliquer. Voyez les conditions d'utilisation pour plus de détails, ainsi que les crédits graphiques. En cas de réutilisation des textes de cette page, voyez comment citer les auteurs et mentionner la licence.
Wikipedia® est une marque déposée de la Wikimedia Foundation, Inc., organisation de bienfaisance régie par le paragraphe 501(c)(3) du code fiscal des États-Unis.

Politique de confidentialité À propos de Wikipédia Avertissements Développeurs Version mobile